



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«МИРЭА – Российский технологический университет»

РТУ МИРЭА

**Институт информационных технологий (ИИТ)
Кафедра цифровой трансформации (ЦТ)**

ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ
по дисциплине «Разработка баз данных»

Практическое занятие № 1

Студенты группы *ИКБО-42-23 Голев С.С.*

(подпись)

Ассистент *Морозов Д.В.*

(подпись)

Отчет представлен «__» _____ 2025 г.

Москва 2025 г.

СОДЕРЖАНИЕ

СОДЕРЖАНИЕ.....	2
ПОСТАНОВКА ЗАДАЧИ.....	3
ХОД ВЫПОЛНЕНИЯ РАБОТЫ.....	4
Создание структуры данных.....	4
Элементы списка выборки – SELECT.....	6
Составление запросов на выборку.....	9

ПОСТАНОВКА ЗАДАЧИ

Для выполнения практической работы необходимо последовательно выполнить следующие шаги, основываясь на логической модели данных, которая была спроектирована в рамках курса «Проектирование баз данных» в предыдущем семестре:

1. На основе логической модели данных, созданной в прошлом семестре, письменно описать не менее 5 различных бизнес-правил и не менее 3 ограничений целостности для таблиц. Выбор бизнес-правил и ограничений целостности производится на усмотрение студента. Результаты представить в виде таблицы.

2. С использованием DDL-оператора CREATE TABLE создать все необходимые таблицы (согласно созданной в прошлом семестре логической модели данных) в СУБД Postgres Pro, корректно реализовав все описанные ограничения целостности.

3. Заполнить созданные таблицы согласованными тестовыми данными (не менее 5-7 записей на таблицу, где это применимо) с помощью оператора INSERT INTO.

4. Составить и выполнить не менее 6 SQL-запросов к таблицам, иллюстрирующих использование различных операторов SELECT и WHERE, согласно перечню, указанному в задании (см. Ход выполнения работы). В составленных запросах должны быть использованы все приведённые примеры – .

5. Составить и выполнить по два SQL-запроса к таблицам для демонстрации работы предложений ORDER BY, GROUP BY и HAVING.

6. Каждый SQL-запрос сопровождать комментарием, объясняющим его назначение и логику работы.

ХОД ВЫПОЛНЕНИЯ РАБОТЫ

Для таблицы ordering из логической модели, составим описание атрибутов и ограничений.

Таблица 1 – Описание ограничений для таблицы ordering

Название	Тип данных	Ограничение	Бизнес-правило
id	INT	PRIMARY	Уникальный идентификатор заказа
total	TEXT	NOT NULL, CHECK (>0)	Обязательная цена не равная нулю
date	INT	NOT NULL, CHECK (date >= CURRENT_DATE)	Дата доставки продукта, не может быть в прошлом
status	TEXT	NOT NULL	Обязательное наличие статуса заказа
delivery_method	INT	NOT NULL	Обязательный метод заказа

Создание структуры данных

С помощью оператора CREATE TABLE создадим таблицу основываясь на логической модели.

```
<postgres> ... <postgres> L... <postgres> S... <postgres> S... <postgres> S... <postgres> ... <postgres> H... <postgres> H...
● CREATE TABLE position (
  id_position INTEGER NOT NULL PRIMARY KEY,
  title TEXT NOT NULL,
  salary REAL NOT NULL
);

● CREATE TABLE supplier (
  id_supplier INTEGER NOT NULL PRIMARY KEY,
  org_title TEXT NOT NULL,
  c_address TEXT NOT NULL,
  phone TEXT NOT NULL
);

● CREATE TABLE category_of_rtl (
  id_category_of_rtl INTEGER NOT NULL PRIMARY KEY,
  description TEXT
);

● CREATE TABLE access_data (
  id_access_data INTEGER NOT NULL PRIMARY KEY,
  id_client INTEGER NOT NULL,
  login TEXT NOT NULL,
  password TEXT NOT NULL,
  FOREIGN KEY (id_client) REFERENCES client (id_client)
);

● CREATE TABLE client (
  id_client INTEGER NOT NULL PRIMARY KEY,
  name TEXT NOT NULL,
  phone TEXT NOT NULL
);

● CREATE TABLE client_address (
  id_address INTEGER NOT NULL PRIMARY KEY,
  id_client INTEGER NOT NULL,
  c_address TEXT NOT NULL,
  FOREIGN KEY (id_client) REFERENCES client (id_client)
);

● CREATE TABLE employee (
  id_employee INTEGER NOT NULL PRIMARY KEY,
  id_position INTEGER NOT NULL,
  name TEXT NOT NULL,
```

Рисунок 1 – Скрипт для создания таблиц

Заполним эту таблицу данными с помощью оператора INSERT INTO.

```
<postgres> ... <postgres> L... <postgres> S... <postgres> S... <postgres> S... <postgres> ... <postgres> H... <postgres> H...
● INSERT INTO position (id_position, title, salary) VALUES
(401, 'Support Agent', 2800.00),
(402, 'Courier', 2400.00),
(403, 'Warehouse Ops', 3000.00),
(404, 'Sales Manager', 4200.00),
(405, 'Accountant', 3800.00),
(406, 'Engineer', 5000.00);

● INSERT INTO supplier (id_supplier, org_title, c_address, phone) VALUES
(11, 'Circuit Solutions Ltd.', '789 Tech Park, Silicon City', '+12025550123'),
(21, 'Hardware Innovations Inc.', '456 Industrial Ave, Tech Town', '+13035550124'),
(31, 'Microchip Enterprises', '101 Chip Lane, Circuit City', '+14045550125'),
(41, 'Digital Designs Co.', '321 Processor Rd, Data Valley', '+15055550126'),
(51, 'Tech Components LLC', '654 Module St, Innovation Hub', '+16065550127'),
(61, 'NextGen Semiconductors', '88 Wafer Blvd, Silicon City', '+17075550128');

● INSERT INTO category_of_rtl (id_category_of_rtl, description) VALUES
(901, 'Sensors & IoT'),
(902, 'Smart Home Hubs'),
(903, 'Lighting'),
(904, 'Cameras & Security'),
(905, 'Networking'),
(906, 'Development Boards');

● INSERT INTO client (id_client, name, phone) VALUES
(201, 'Alice Johnson', '+358401112233'),
(202, 'Boris Petrov', '+79261112233'),
(203, 'Carla Gómez', '+34911122334'),
(204, 'David Chen', '+14155552671'),
(205, 'Elena Rossi', '+39021234567'),
(206, 'Farid Khan', '+971501234567');

● INSERT INTO access_data (id_access_data, id_client, login, password) VALUES
(1001, 201, '2001001', '9001001'),
(1002, 202, '2001002', '9001002'),
(1003, 203, '2001003', '9001003'),
(1004, 204, '2001004', '9001004'),
(1005, 205, '2001005', '9001005'),
(1006, 206, '2001006', '9001006');

● INSERT INTO client_address (id_address, id_client, c_address) VALUES
(301, 201, '221B Baker St, London'),
(302, 202, '1600 Amphitheatre Pkwy, Mountain View'),
```

Рисунок 2 – Скрипт для заполнения таблиц

Элементы списка выборки – SELECT

Напишем скрипты с помощью оператора `SELECT` для разных сценариев.

Рисунок 3 – SELECT скрипт для вывода всего содержимого таблицы.

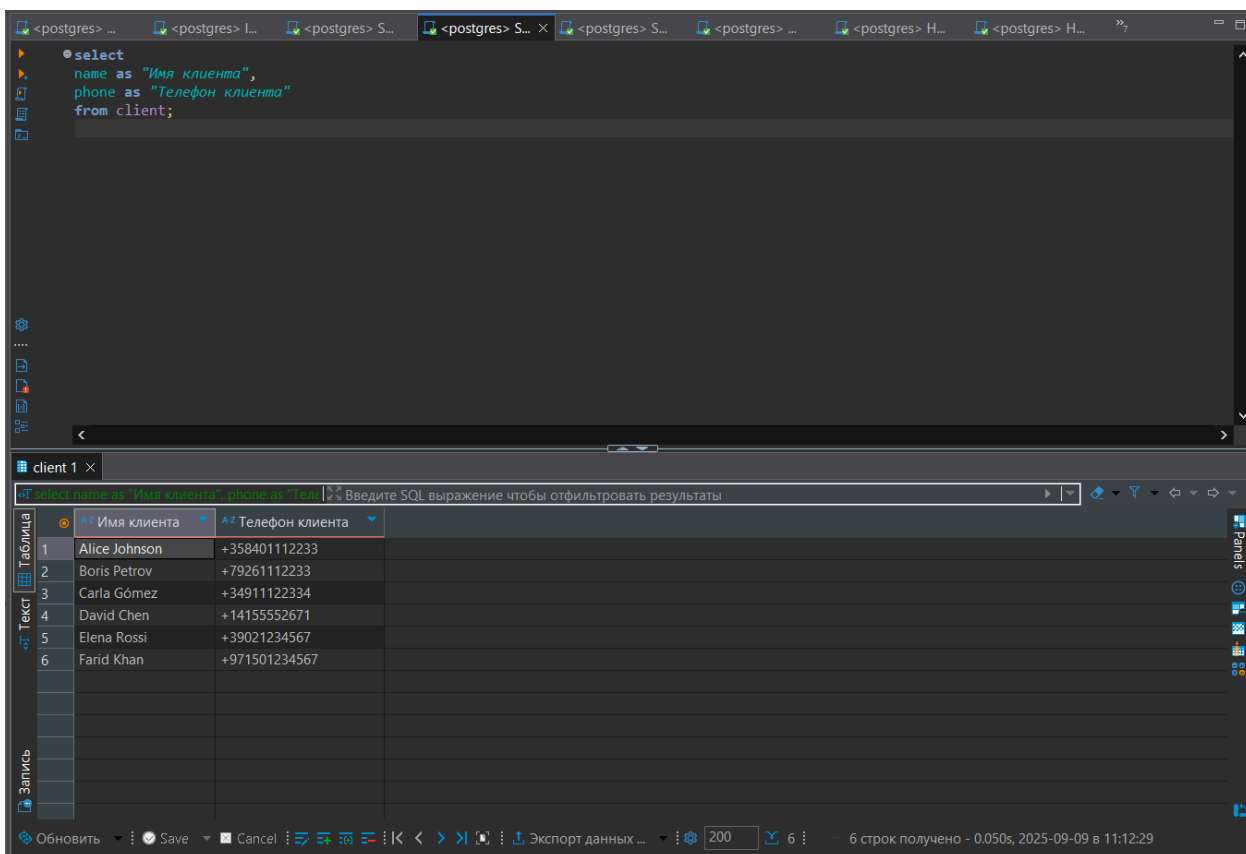


Рисунок 4 – SELECT скрипт для вывода конкретных полей таблицы.

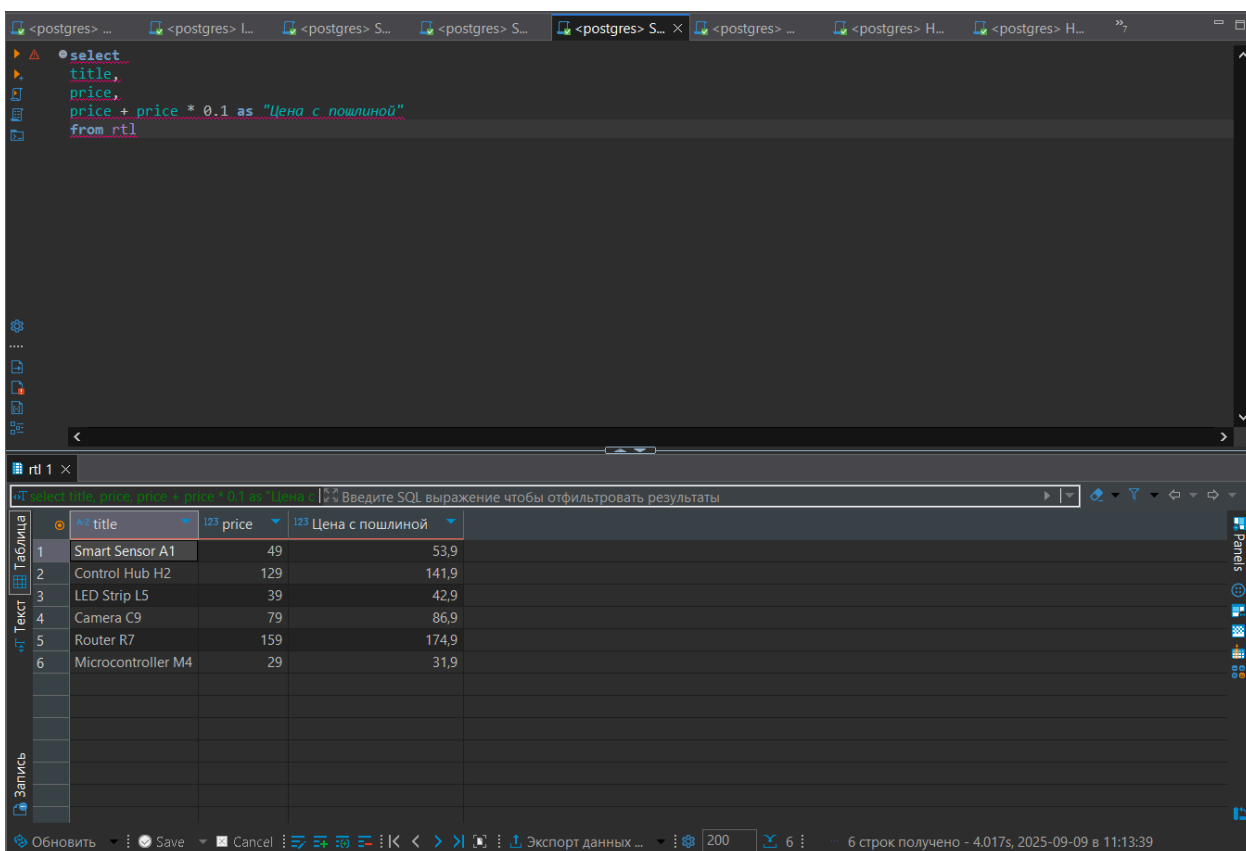


Рисунок 5 – SELECT скрипт для вывода конкретных полей таблицы с использованием математических операций.

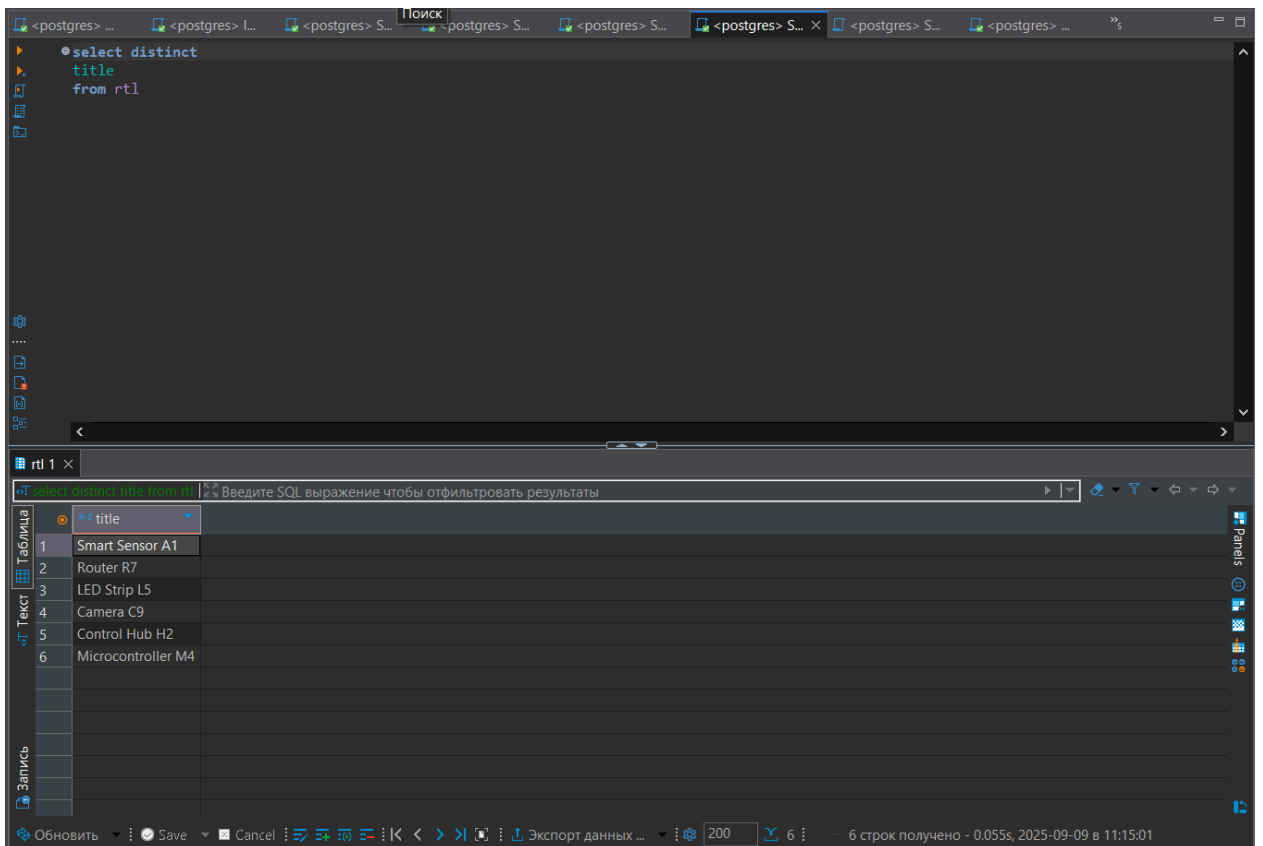


Рисунок 6 – SELECT скрипт для вывода уникальных полей таблицы.

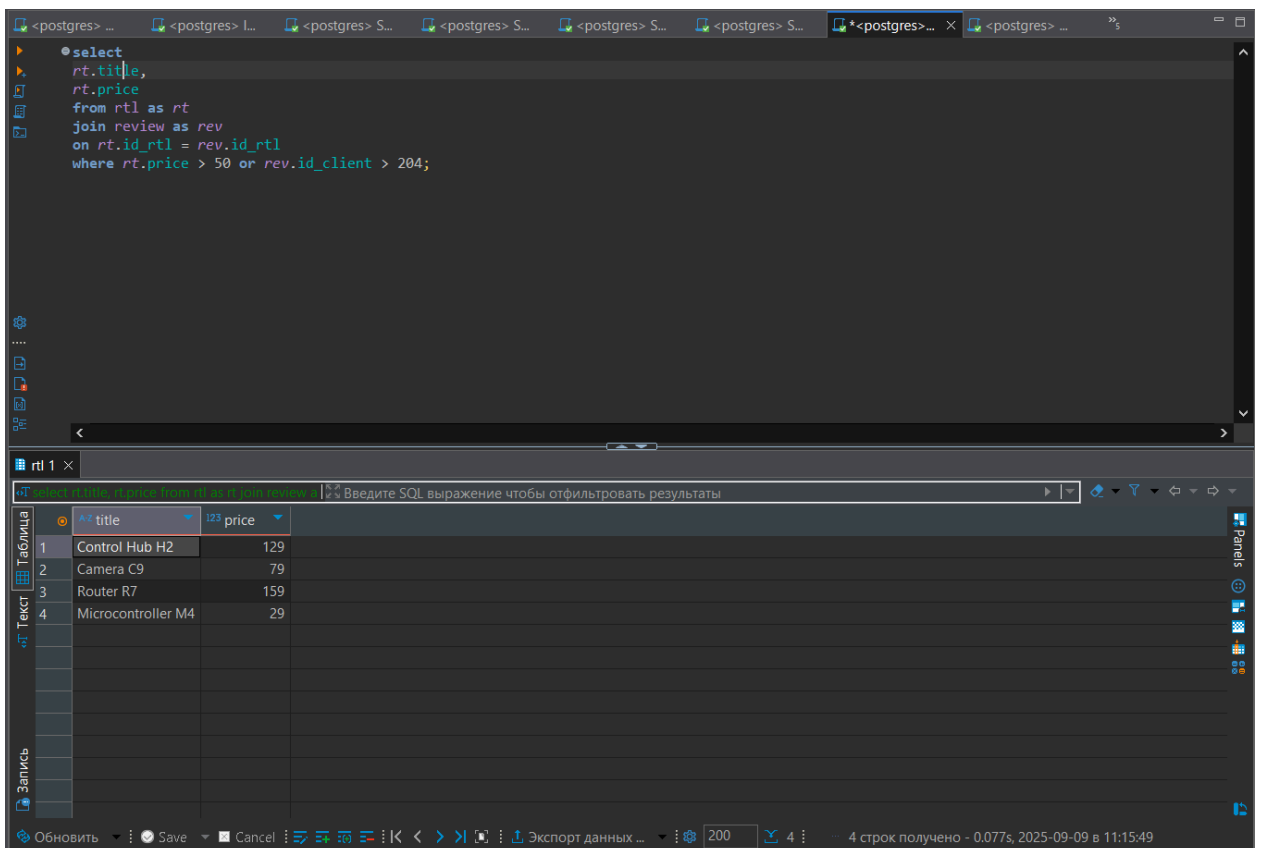


Рисунок 7 – SELECT скрипт для вывода полей таблицы в которых присутствует 'er'.

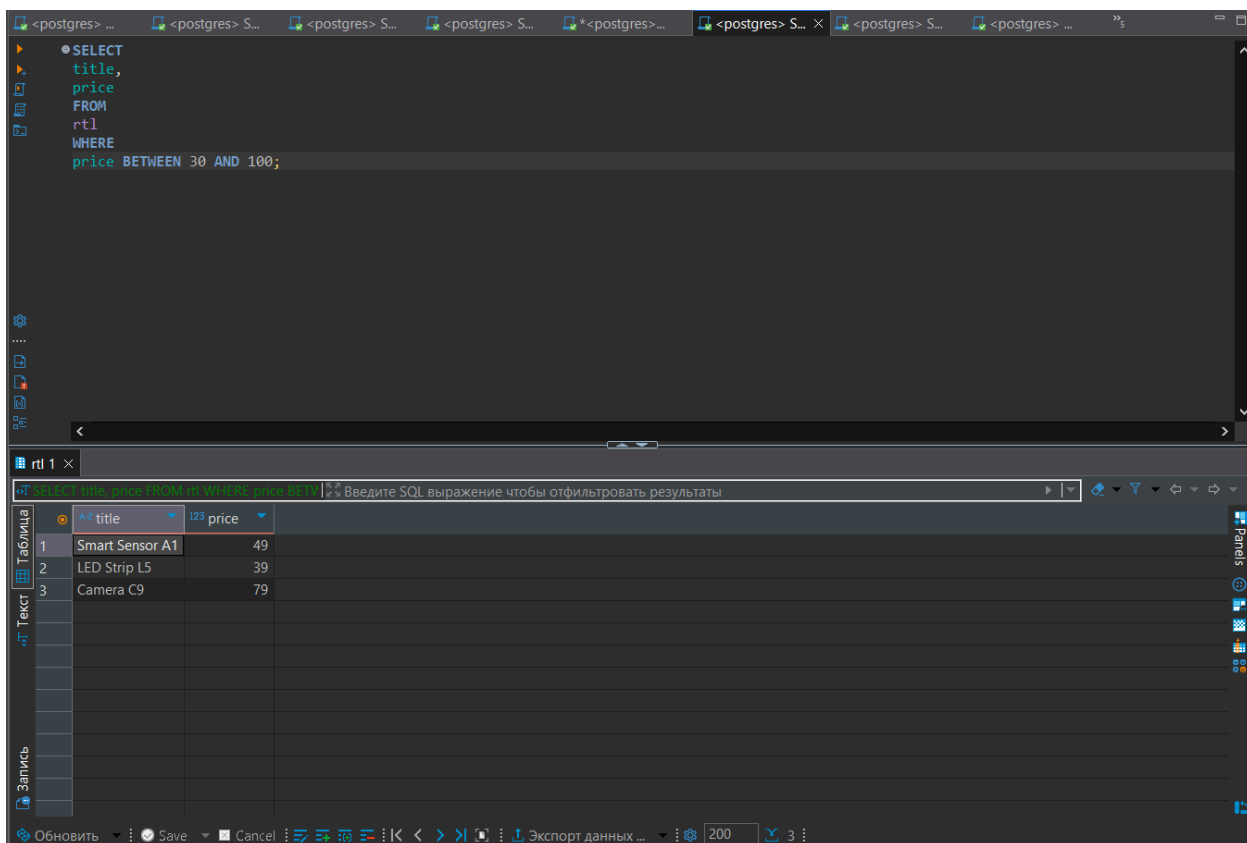


Рисунок 8 – SELECT скрипт для вывода полей таблицы соответствующих выбранному диапазону.

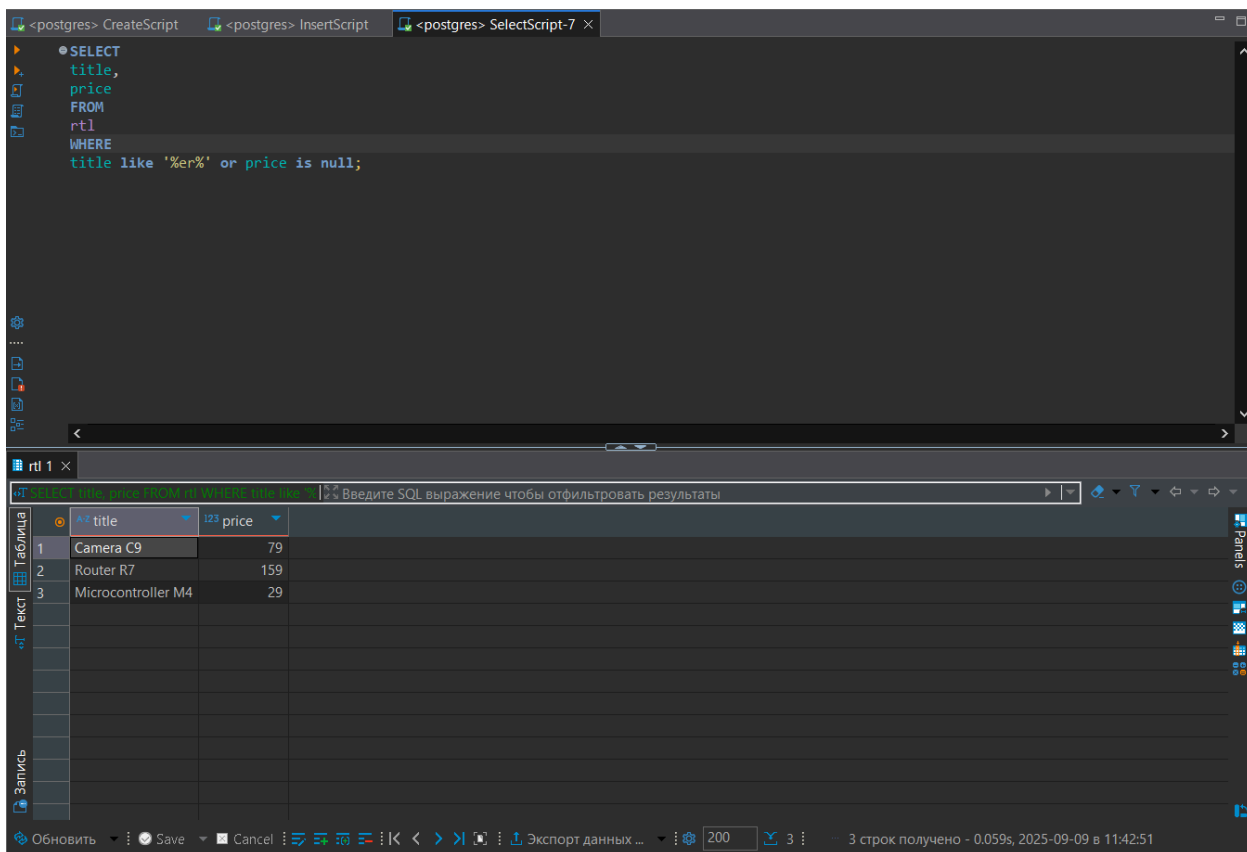


Рисунок 8.1 – SELECT скрипт для вывода полей таблицы с помощью операторов LIKE и IS NULL.

Составление запросов на выборку

Напишем скрипты используя операторы ий ORDER BY, GROUP BY и HAVING.

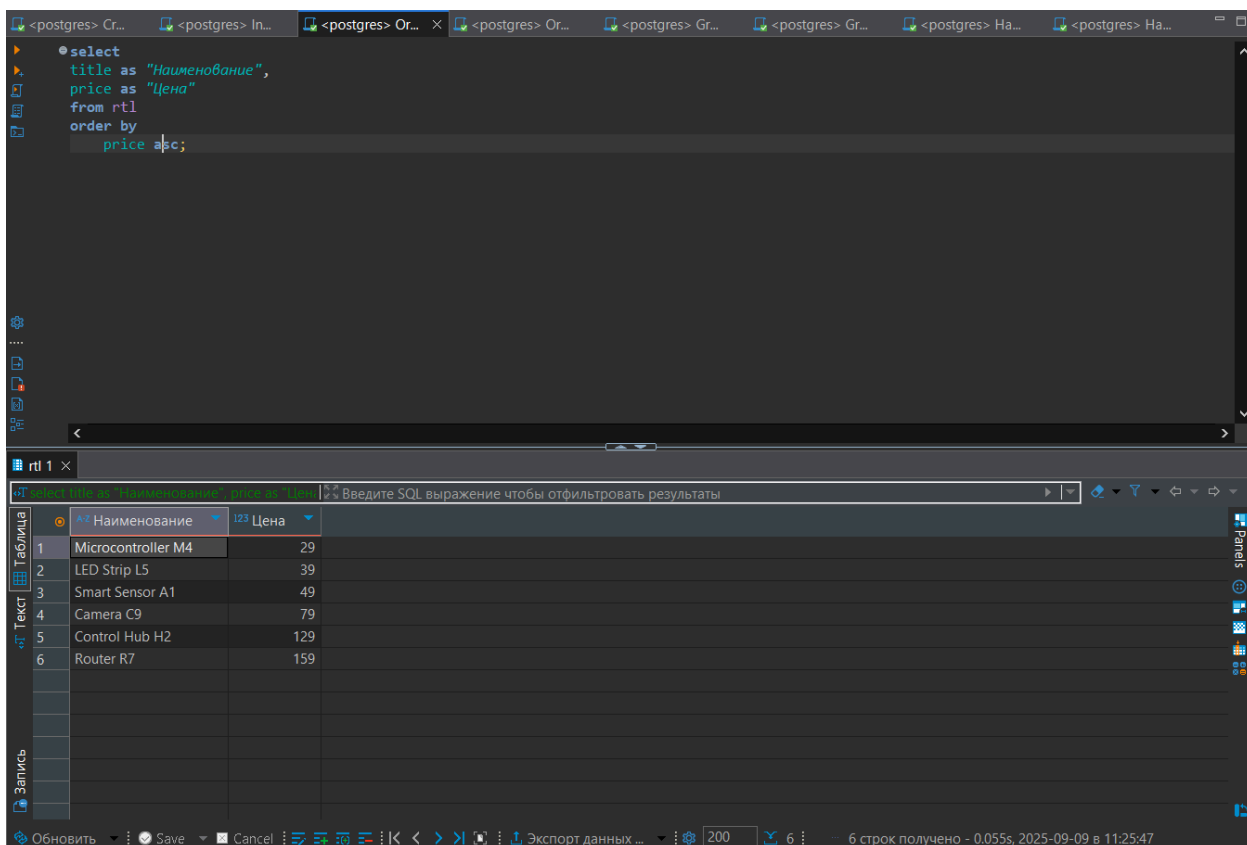


Рисунок 9 – ORDER BY скрипт с использованием оператора ASC.

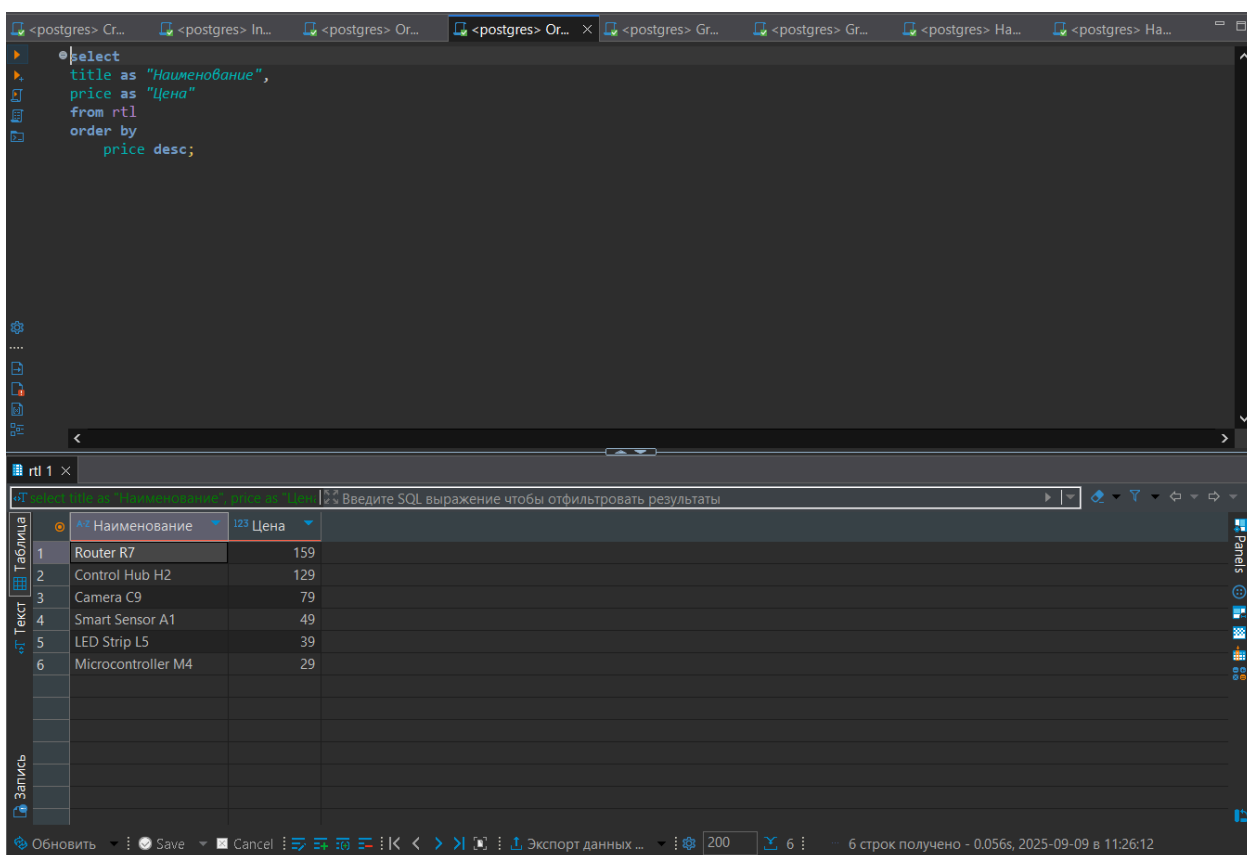


Рисунок 10 – ORDER BY скрипт с использованием оператора DESC.

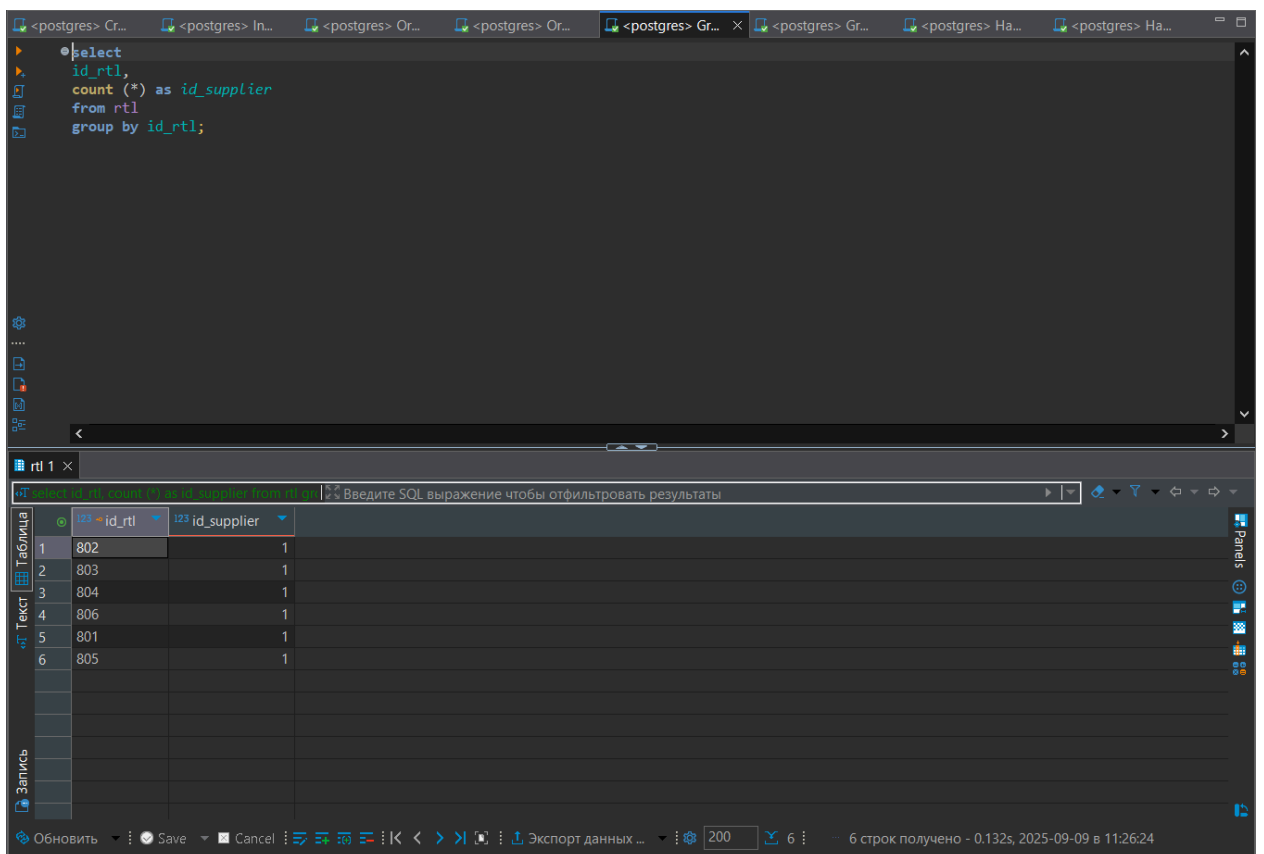


Рисунок 11 – GROUP BY скрипт с использованием оператора COUNT.

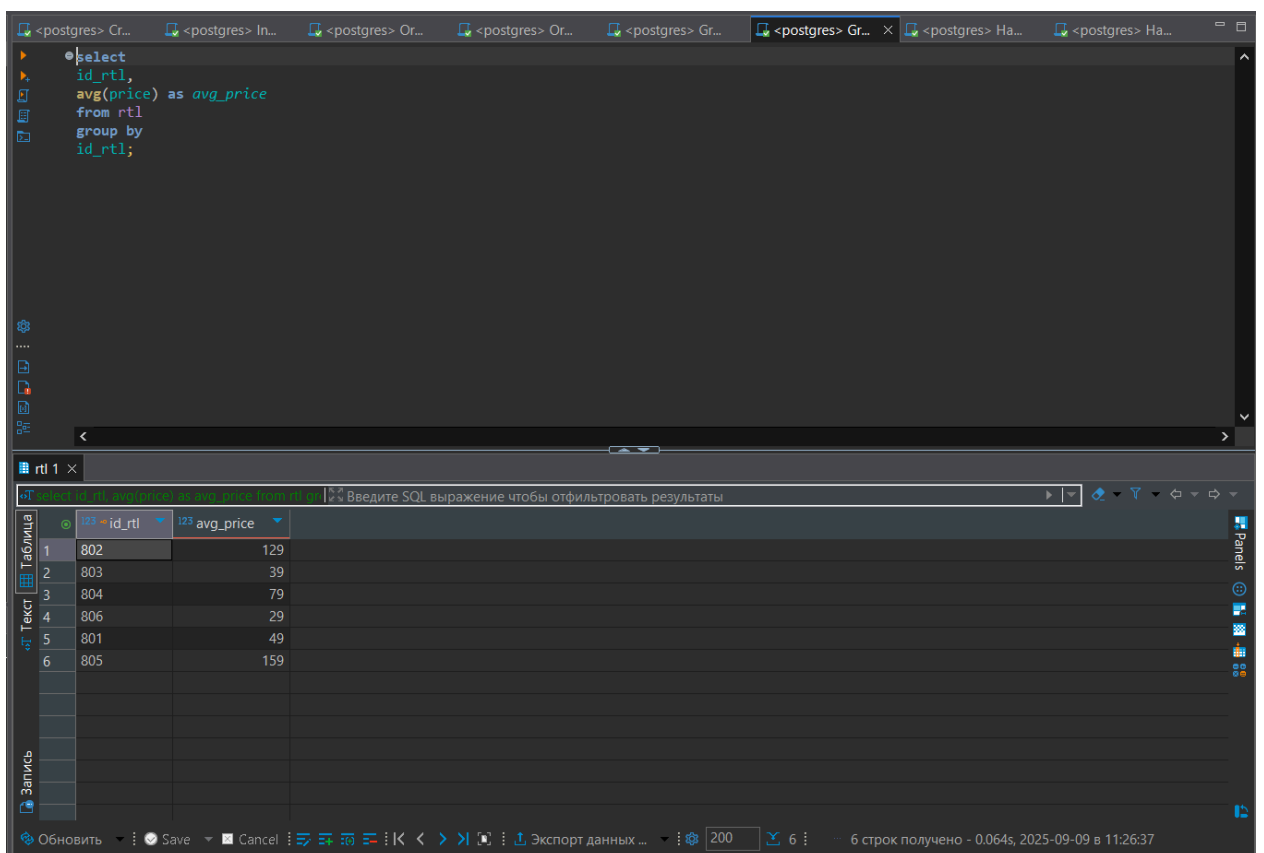


Рисунок 11 – GROUP BY скрипт с использованием оператора AVG.

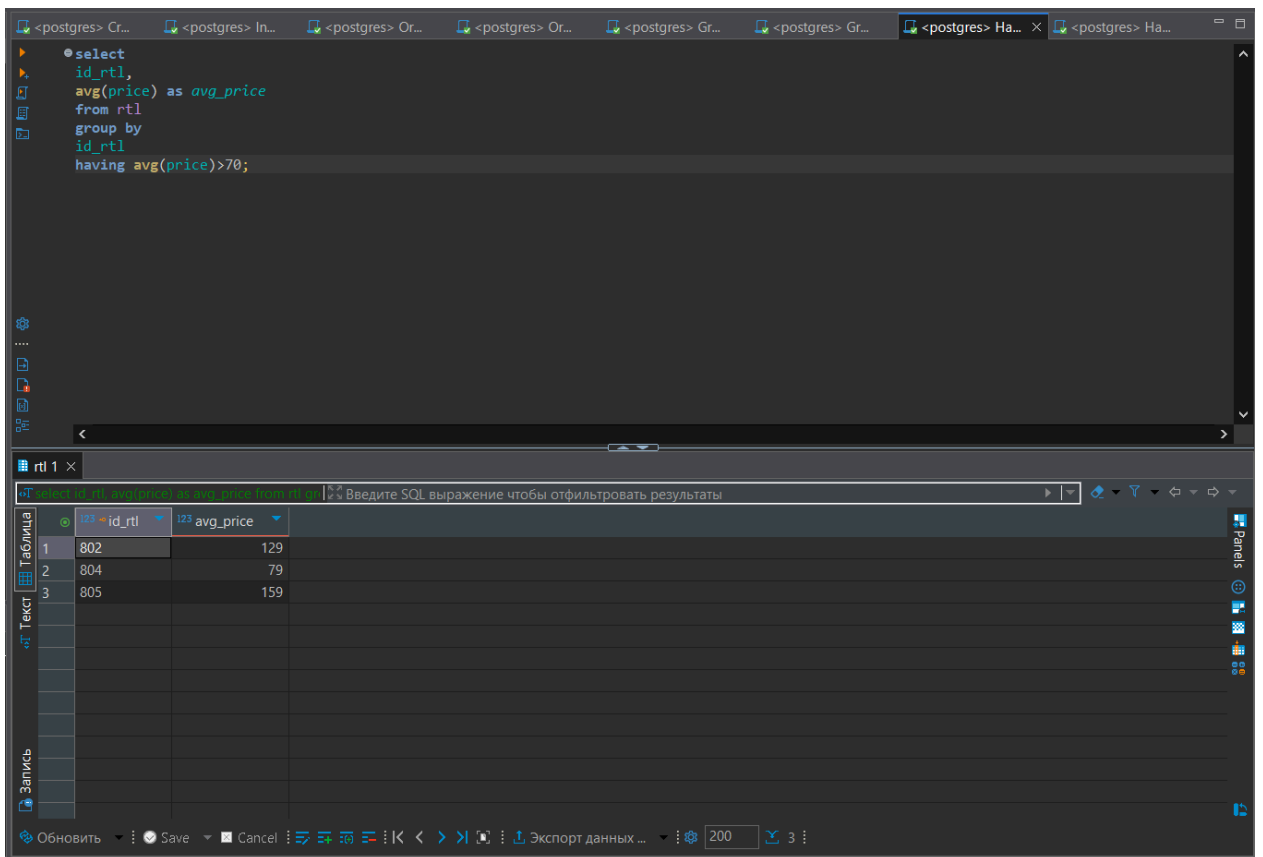


Рисунок 12 – HAVING скрипт с использованием оператора AVG.

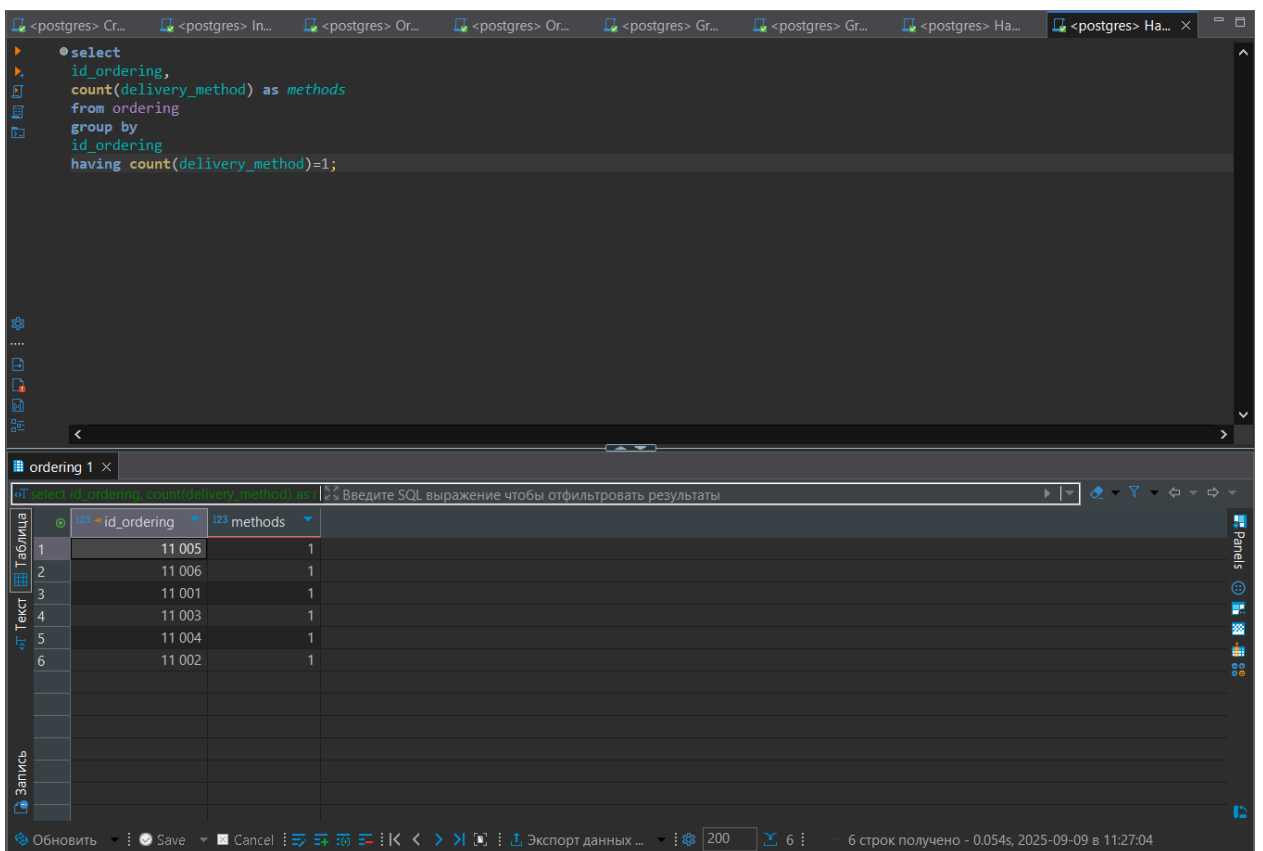


Рисунок 13 – HAVING скрипт с использованием оператора COUNT.